FAIRification of lab data

ISC proposal

Martin Schobben

2021-10-28

Contents

1 Signatories

1.1 Project team

Martin Schobben, FAIReLABS, Utrecht, the Netherlands
Janou Koskamp, Utrecht University, Utrecht, the Netherlands
Johan Renaudie, Museum für Naturkunde - Leibniz Institute for Evolution and Biodiversity Science, Berlin, Germany
Terrell Russell, iRODS Consortium, Chapel Hill, United States of America

A detailed account of the individual team member's interests and skills can be found in Section 5.1.

1.2 Consulted

Bas van de Schootbrugge, Utrecht University, Utrecht, the Netherlands
Francien Peterse, Utrecht University, Utrecht, the Netherlands
Inigo Müller, University of Geneva, Geneva, Switzerland
Jan Voskuil, Taxonic & Ontologist, The Hague, the Netherlands
Lubos Polerecky, Utrecht University, Utrecht, the Netherlands
Mariette Wolthers, Utrecht University, Utrecht, the Netherlands
Nicole Geerlings, College Hageveld, Heemstede, the Netherlands
Peter Bijl, Utrecht University, Utrecht, the Netherlands
William Foster, University Hamburg, Hamburg, Germany
From the R Consortium's ICS: Hadley Wickham

2 The Problem

Data from analytical laboratories is omnipresent in our daily lives from COVID-19 infections, meteorology, forensics, and the quality of our drinking water. Unfortunately, laboratory data streams are often fragmented and not well curated. We reason that this is caused by the range of analytical instruments populating the lab—each with their own closed-sourced vendor-supplied data models and software suites for subsequent data processing, analysing, and diagnostics (see "Unconnected Lab" Fig 1). These various data models stored on local devices, if accessible at all, are not easily integrated in a centralised data management infrastructure with sufficient data description, e.g., provenance and quality assurance (metadata). This so-called "vendor lock-in" further prevents transparency of the workflow from raw to analysed data. Although low-level access to raw data and insights in workflows is not necessary for all researchers/data analyst, it can be important for special purpose research questions, possibly sparking new innovations and discoveries. The fragmented and partly obscured nature of data streams from analytical laboratories therefore conflicts with data management principles, such as formalised in the Findable, Accessible, Interoperable, and Reusable (FAIR) data guiding principles (Wilkinson et al. 2016), and have a negative impact on the reproducibility of science. Existing solutions for reading data, such as *readr* (Wickham and Hester 2021) and *vroom* (Hester and Wickham 2021), can be cumbersome for this particular task, as the unstructured and large (>1,000 lines) (meta)data formats prevents straightforward parsing. This has resulted in a series of custom solutions, e.g., *xrftools* (Dunnington 2021), *isoreader* (Kopf, Davidheiser-Kroll, and Kocken 2021), and *point* (Schobben 2021), for various machine-specific data models (this is a non-exhaustive list).

Hence, a more universal solution to this problem of analytical data collection and harmonisation is therefore a rewarding endeavour for future innovations and discoveries. In addition, FAIR data is conducive to an inclusive, connected worldwide academic community—providing opportunities for developing countries that do not have the same resources for data generation as wealthy countries.

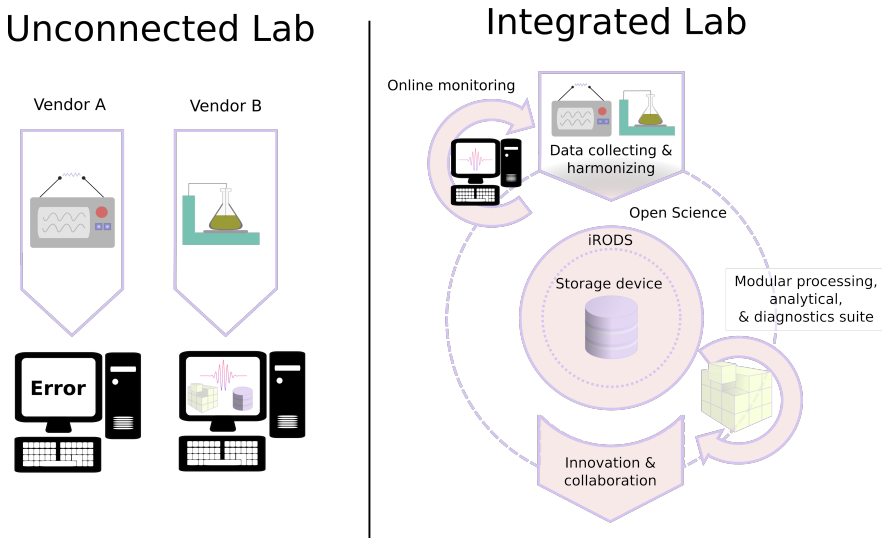

Figure 1: Integrated lab solution versus traditional unconnected lab set-ups. iRODS = Integrated Rule-Oriented Data System.

3 The proposal

3.1 Overview

The integrated lab is a solution to centralise data management of the traditional unconnected lab (see "Integrated Lab" Fig 1). A first step in the realisation of an integrated lab would encompass a solution for collecting and harmonising data streams from various lab instruments. The development of the R package *panacea*[1] will attempt to provide a more universal solution for parsing unstructured (meta)data formats in a rectangular format—notably, separating variables, units, and values. This solution would therefore make analytical data more easily accessible for both humans and machines. In extension we intend that this solution centralises data management of labs by facilitating automatic data ingest (i.e., data import) as a subsystem of iRODS (Integrated Rule-Oriented Data System) (Rajasekar et al. 2010, 2015).

Besides addressing the vendor lock-in of analytical data and optimized data management solutions, this tool has several other benefits:

1. New software updates of the vendor-supplied software that impact the output format can be more easily accommodated, and do not require cumbersome updates of custom R solutions.
2. Data formats from defunct software and vendors can be more conveniently analysed and/or archived in a central data management system.
3. The integration of (meta)data from different sources can aid online monitoring of lab performance. For example, centralised data management could theoretically provide opportunities for early detection of problems, such as sample/reagent pollution and anomalous lab-environmental conditions. The latter problems would be much harder to detect with stand-alone vendor supplied solutions.

To conclude, we want to put scientist back in control of their data without having to rely on closed-sourced vendor software. This could save countless working hours and large sums of taxpayer money. Even so, the envisioned solution might not enjoy the broadest employment by the R community, we hope to open-up a dialogue about the transparency of data life cycles that are a cornerstone of our society. Together with the benefits of integrated labs, this could lead to new innovations, more transparent science, and promote inclusiveness in the academic community and beyond.

[1] Portable ANalytical data Aggregation and Coordination for database Entry and Access

3.2 Detail

Observational data generated by commercial analytical instrumentation and accompanying software is often recorded as unstructured text files.[2] In this context we refer to "unstructured" as incorporating tab-delimited or fix-width tables (Fig. 2) of data intermingled with lines of, one or more, variable-value-unit triplets (see lines 1, 3 and 4 of Fig. 2). On top of that, files often consists of >1,000 lines, and syntactic inconsistencies are not uncommon.

```
 1: 2021-09-20  20:15                                Sample ID: MON-233
 2:
 3: Peak Height Distribution: 210 V, EMHV: 2350 mV
 4: Position: x=12um; y=2um; z=100um
 5:
 6:   Time (s)  Count
 7:   1         56
 8:   2         60
 9:   3         64
10:   4         64
11:   5         57
12:   6         59
13:   7         58
14:   8         58
15:   9         62
16:   10        54
```

Figure 2: An excerpt of how unstructured raw data files from analytical laboratory equipment typically looks like. This is an imaginary excerpt modelled after the main applicants experience with this type of data output. Note, that this is still a fairly structured data format in respect to what one can find in the wild.

This lack of structure is perceivably less dramatic than that encountered for information entrained in emails, Twitter feeds and literature. Nonetheless, the primary task of identifying variables, values, and units, as distinct entities as well as larger structures (e.g., tables), is the most challenging task in this undertaking.

We envision three possible solutions, which require varying degrees of human intervention (Table 1).

1. A mechanism to aid the location of variables based on user input.
2. A human-crafted (and adaptable) rule based system.
3. A natural language processing approach involving self-supervised machine learning.

Solution #1 requires the input of variable names and their context (i.e., a table or line), whereby regular expression locate the respective variables for subsequent parsing. This approach would thus require considerable knowledge of the end-user considering the raw data and its internal organization, and is only a slight deviation of widely popular packages such as *readr* (Wickham and Hester 2021) and *vroom* (Hester and Wickham 2021). It is therefore also the most feasible of the proposed solutions.

The next two solutions would be preceded by a step entailing text normalization through tokenization. Tokenization will be performed with cascades of regular expressions for word (entity) delimiters. These delimiters will likely not be based on word boundaries, but instead use a combination of punctuations and tabs as delimiters. On the other hand, special character and alphanumeric combinations, as occur in paths and dates, should constitute one token, and require special consideration.

Solution #2 would require writing a set of more-or-less universal rules that describe typical formatting structures of analytical instrument output. After preprocessing, we suspect that it is possible to generalise that all numeric tokens (strings) can be tagged as values. In turn, frequencies of the tokens in a collection of files can then help separate the remaining non-numeric values from the variables and units. Finally, a set of

[2]Note, that the methods proposed here still require a vendor-supplied electrical-to-digital signal conversion

Table 1: The required human intervention to parse unstructured data for each of the proposed solutions and the perceived risk of developing the associated solution.

Solutions	Human-action	Risk
#1	high	low
#3	medium	medium
#3	low	high

Table 2: Provisional pancaea return value, based on the analytical data output of a virtual machine of Fig. 1

Variable	Unit	Relation	Values
Date Time	NA	file 1 , line 1 , section 1	2021-09-20 20:15:00
Sample ID	NA	file 1 , line 1 , section 2	MON-233
Peak Height Distribution	V	file 1 , line 3 , section 1	210
EMHV	mV	file 1 , line 3 , section 2	2350
Position-x	um	file 1 , line 4 , section 1	12
Position-y	um	file 1 , line 4 , section 2	2
Position-z	um	file 1 , line 4 , section 3	100
Time	s	file 1 , table 1 , column 1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Count	NA	file 1 , table 1 , column 2	56, 60, 64, 64, 57, 59, 58, 58, 62, 54

rules based on sentence boundaries, punctuation, and delimiters might help recognize larger structures (e.g., tables) that can help tie together the variables and their constituent units and variable values.

Solution #3 would be almost free of human intervention. This method could be reminiscent of part-of-speech tagging in order to recognise the individual entities of the triplet; variables, values, and units. Recognition of larger structures (i.e, tables) might be based on chunking approaches that reminisce the methods serving context free grammar and/or dependency grammar solutions in NLP (Jurafsky and Martin 2021).

Ultimately, the (meta)data tagging solution(s) will form the engine of the to-be-developed core function of *panacea*. This function for the read-out of the instrument data will then proceed with parsing of unstructured data into a more convenient human and machine-readable format. This output is preliminary envisioned to constitute a *tibble* (Müller and Wickham 2021) with columns; variable (of type character), unit (of type character), relation (of type list), which constitutes a network of relations describing structures in the original document, and values (of type list) (see Table 2). The user-interface of the function will be modelled after *readr* (Wickham and Hester 2021) and *vroom* (Hester and Wickham 2021).

Based on a twofold reasoning, we propose encoding this solution in the C++ language. Firstly, we want to ensure compatibility with external data management software, notably iRODS. In this use-case, the compiled C++ source code of *panacea* could be adapted to create a standardized protocol for ingestion into a central data management system. The R package *rirods* (Chytracek et al. 2015) will be used to query the iRODS API. This auxiliary package will, however, require some maintenance and adaptations. The second consideration is performance related, e.g., the demanding operation of tokenizing a large corpus. This approach of extending the R core interpreter with C++ ensures a lean and fast approach. In addition, the usage of the R package *cpp11* (Hester 2021) enables the ALTREP framework for lazy load of data in R, ensuring further speed and convenience of the functionality.

4 Project plan

4.1 Start-up phase

The development of *panacea* is a central part of a newly initiated consortium; FAIReLABS[3], dedicated to researching and developing solutions to make laboratory data, throughout the whole cycle from generation to analysing, more transparent, accessible, and customisable. Thus FAIReLABS, and the here proposed package, is meant to be conducive to innovation within an analytical laboratory environment, and foster inclusiveness through open science. Besides research and development, it is intended that FAIReLABS provides courses/workshops in data management practices and reproducible science as well as consultation in facilitating the transition to an integrated lab (see Fig. 1). The initial step is already undertaken by starting a new GitHub Organisation for FAIReLABS, which also host this proposal as a public repository. In turn, package development encompasses soliciting specific use-cases from the R community and laboratory facilities. A close collaboration with the Department of Earth Sciences, Utrecht University (UU), the Netherlands, and their analytical laboratory infrastructure, is already foreseen. Nonetheless, the development will benefit from having a good overview of the types of data and data models produced by analytical equipment in a range of laboratories. We will opt for an MIT license and a code of conduct, which will follow the Contributor Covenant guide lines. Combined this ensures that contributions to the package can be done in a safe, inclusive, welcoming, and harassment-free environment conducive for collaborative package development, and ensuring down-stream re-usage of the developed software. Reporting of the progress of the project to both (lab-)users and developers will help ensure that we stay on track and thus develop a solution that has a broad future implementation.

4.2 Technical delivery

The duration of the project will be 12 months. The "**deliverable**" gives a convenient measure of project's progress.

Months 1–2

- Documentation of use-cases combined with on-premise visits to lab facilities.
- Determine the feasibility of the solutions for the engine of the core function as discussed in section (3.2) and make a selection.
- **Deliverable:** We report our finding on the current state of data management infrastructures and common data models (i.e., instrument output) in analytical laboratory settings as a blog post.

Months 3–4

- Start with basic package set-up with *devtools* (Wickham, Hester, and Chang 2021), create source scripts in C++ for loading data files.
- Follow best practices from the start of package development; e.g., documenting progress, maintaining a functioning Git master branch and usage of development branches for experimental updates. This will be published on GitHub from the start, and tags are created when milestones are hit to benefit progress tracking. In addition, unit tests are constantly developed to ensure that a particular behaviour of a function is, and remains, correct (and also regularly checking code coverage of said tests). Lastly, we test code, portability, and documentation with `R CMD check` and with continuous integration provided by Travis CI.
- **Deliverable:** A GitHub repo with the basis of the package.

Months 5–6

[3]FAIR refers to the guiding principles for data: Findable, Accessible, Interoperable, and Reusable (Wilkinson et al. 2016)

- Minimal functionality to locate variables within their context (Solution 1), and test on different use-cases.
- **Deliverable:** A tag in the GitHub repo annotating the milestone of a functioning (meta)data tagging solution.

Months 7–8

- Include iRODS functionality.
- Present package at conference(s) targeting users (natural science conference) and/or developers (R or open science conference).
- **Deliverable:** Installable package on GitHub with documentation as vignettes and website with *pkgdown* (Wickham and Hesselberth 2020).

Months 9–10

- Test use cases for the integrated lab and using C++ code basis for centralised data management (iRODS) in a laboratory setting (UU).
- Optional: Further development on the engine for data collecting and harmonization, e.g., probing solutions 2 and 3.
- **Deliverable:** A tag in the GitHub repo annotating the milestone of successful integration with iRODS.

Months 11–12

- Publish on CRAN.
- Include usage of package in teaching/course material provided by FAIReLABS.
- Optional: Further development on the engine for data collecting and harmonization, e.g., probing solutions 2 and 3.
- **Deliverable:** Installable package on CRAN.

4.3 Other aspects

We will garner attention on the problem of unconnected labs and their bearing on open science, and our proposed solutions, through several channels (see also the timeline above). Firstly, we intend to describe the problem in more detail by gathering more insight from specific laboratory settings in a dedicated blog post at the start of the project. The former post also proposes strategies to tackle this problem, thereby setting the stage for a collaborative platform for the development of the package. Besides being an integral aspect for future development of courses and consultation delivered by the FAIReLABS organisation (see above), we actively seek to advertise the product by presenting our finding at conferences; either user-specific (natural science conference) and/or the developers community (e.g. R or open science conference).

5 Requirements

The realisation of this package requires a collaborative environment that includes the potential users, and their specific requirements for processing analytical data, as well as developers and data scientist with expertise in a range of disciplines. In regards to development, we brought together a multi-disciplinary team, and consulted experts of data management, machine learning, and the integration of C++ and R.

5.1 People

The project team will try to form a comprehensive picture of the current state of data management practices in laboratories through direct interaction with lab-users. In addition, they take control in all steps of development, documentation and outreach of the package. Dedicated consultants have been contacted and their expertise is regarded as an essential aid for successful deployment of the plan.

The project lead (MS) is an Earth scientist with 10 years of experience in academic research, and he has worked in several analytical laboratory facilities (MfN Berlin, University of Leeds, and Utrecht University). He also has a solid basis in data-analysis and programming with R, and has started developing packages for analysing isotope chemical data (see point). Teaching and helping others to encode R solutions has been another of his passions, such as the development of workshops, and by founding of an R help desk at the UU (uu-code-club).

JK, a computational chemist, has 5 years of experience in computer simulations, such as Molecular Dynamics, Umbrella Sampling and metadynamics. She is currently working as a postdoc. Previously, she worked in an analytical laboratory at the R&D department of Canon. In both jobs she used different programs (matlab, python and fortran) to process large amounts of data.

JR, also a geoscientist, has expertise in data management (being the main maintainer and developer of Neptune; one of the largest paleontology database), data analysis (primarily in R and python), machine learning (see e. g. a CNN-based radiolarian classifier) and scientific software development (see e. g. NSB_ADP_wx or Raritas; two pieces of software designed in particular for increasing data reproducibility and reusability in paleontology and stratigraphy). JR was also the organizer of a programming club at the MfN (Mfn Code Clinic) from 2015 to 2018.

TR oversees the iRODS development team and handles code review, package management, documentation, and high level architecture design. He's interested in distributed systems, metadata, security, and open source software that accelerates science. TR holds a Ph.D. in Information Science from the University of North Carolina at Chapel Hill and has been working on iRODS since 2008. In his current role, he also provides management and oversight of the iRODS Consortium.

5.2 Processes

A prime controller in the initiation of the project is the report (and blog post) in the first two months (see, deliverable Months 1–2; Section 4.2), which tries to give an overview of existing data management infrastructures and common data models (i.e., instrument output) in analytical laboratory settings. Based on this deliverable, and input from lab-users, adoptions to the initial plan can be made. Specifically, it helps select what solution should be adopted for data selection and harmonisation. To foster an efficient start-up and continues collaboration, we adopt a strategy of publishing advancements in development at an early stage, so that testing and evaluation can begin as soon as possible. Feedback on these early developments is sought actively through our dedicated list of consultants, but also the community at large through Twitter and other channels. Throughout this process, we will make sure that the code of conduct, as outlined in Section 4.1, is adhered to.

5.3 Tools & Tech

For successful delivery of the package we need access to large quantities of raw data from various analytical instruments. We have secured access to data from Utrecht University and the MfN Berlin. GitHub is essential for the collaborative character of the work. No additional computing facilities are envisioned at the moment.

Table 3: Itemized budget.

Item	Currency	Price
Travis CI 1 year core plan	$	759
travel expenses for local lab visits	$	200
conferences	$	2,500
Total	$	3,459

5.4 Funding

We request $959 for direct project costs and $2,500 in funding for the attendance of two conference for one person (Table 3). Provisionally, addressing a potential user-base and the open science community at the EGU General Assembly 2022 Vienna and the ICOSRP 2022 Helsinki (or the iRODS User Group Meeting 2022 Leuven), respectively.

5.5 Summary

Support is requested for the development, documentation as well as outreach of the package.

6 Success

6.1 Definition of done

The deliverable "Installable package on CRAN" of Section 4.2 defines achievement of the minimal viable product (Solution 1; Section 3.2). Progress on the implementation of Solutions 2 and/or 3 is seen as a bonus.

6.2 Measuring success

The actual success during the development phase is measured by the number of contributions and the number of laboratories that we can engage with. The success of the developed package is measured by use-cases through download statistics, and for development purposes, by tracking how many packages will integrate this package.

6.3 Future work

Future work in the sense of technical innovation likely entails application to different file types, such as binary files. In addition, we intend to develop a python package with the same scope. Further progress revolves around integration of the package with the services offered by FAIReLABS. It will enable consulting and implementing better resources for data management in analytical laboratories as well as help teaching efforts focussing on data management and reproducible science. In addition, we consider writing a paper concerning the package for the Journal of Open Source Software, and continuously advertise usage of the package by actively engaging with target user-base at conferences and on social media.

6.4 Key risks

One of the key risks in the process of developing the package is the selection of the appropriate solution (as listed in Section 3.2). Hence the early identification of this bottleneck and the formulation of three contingency plans (i.e., the different solutions) will help alleviate these risks to some extend. Problems and

delays in terms of coordinating community feedback (especially desired use-cases) and contributions (mainly solutions as listed above) could stem from a lack of consensus on the specific solution to be adopted. Hence we aim to address this at the earliest stages of the project (Months 1–2). In terms of tools and technology, we foresee the largest problem in access to enough analytical data. Hence we ensured that we have already a substantial set of data available for testing purposes. All the before mentioned risks could increase the time required to develop the product. However, by defining a set of minimal deliverables, we can at least sketch an accurate image of the current state of data management practices in analytical laboratory facilities and develop a road-map on how to improve these infrastructures. It is also foreseen that solution 1 yields a minimal viable product.

References

Chytracek, Radovan, Bernhard Sonderegger, Richard Cote, and Terrell Russell. 2015. "The Rirods Package Enables Access to File Objects in the iRODS Data Broker System from r." https://github.com/irods/irods_client_library_r_cpp/blob/master/DESCRIPTION.

Dunnington, Dewey. 2021. *Xrftools: XRF Tools for r*. https://github.com/paleolimbot/xrftools.

Hester, Jim. 2021. *Cpp11: A c++11 Interface for r's c Interface*. https://CRAN.R-project.org/package=cpp11.

Hester, Jim, and Hadley Wickham. 2021. *Vroom: Read and Write Rectangular Text Data Quickly*. https://CRAN.R-project.org/package=vroom.

Jurafsky, Daniel, and James H. Martin. 2021. *Speech and Language Processing: An introduction to natural language processing, Computational Linguistics, and Speech Recognition*. http://www.cs.colorado.edu/%7B~%7Dmartin/slp.html.

Kopf, Sebastian, Brett Davidheiser-Kroll, and Ilja Kocken. 2021. *Isoreader: Read Stable Isotope Data Files*. https://github.com/isoverse/isoreader.

Müller, Kirill, and Hadley Wickham. 2021. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Rajasekar, Arcot, Reagan Moore, Chien-Yi Hou, Christopher A. Lee, Richard Marciano, Antoine de Torcy, Michael Wan, et al. 2010. "iRODS Primer: Integrated Rule-Oriented Data System." *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2 (1): 1–143. https://doi.org/10.2200/s00233ed1v01y200912icr012.

Rajasekar, Arcot, Terrell Russell, Jason Coposky, Antoine de Torcy, Hao Xu, Michael Wan, Reagan W. Moore, et al. 2015. *The integrated Rule-Oriented Data System (iRODS 3.0) Micro-service Workbook*.

Schobben, Martin. 2021. *Point: Reading, Processing, and Analysing Raw Ion Count Data*. https://martinschobben.github.io/point/.

Wickham, Hadley, and Jay Hesselberth. 2020. *Pkgdown: Make Static HTML Documentation for a Package*. https://CRAN.R-project.org/package=pkgdown.

Wickham, Hadley, and Jim Hester. 2021. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wickham, Hadley, Jim Hester, and Winston Chang. 2021. *Devtools: Tools to Make Developing r Packages Easier*. https://CRAN.R-project.org/package=devtools.

Wilkinson, Mark D, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "Comment: The FAIR Guiding Principles for scientific data management and stewardship." *Scientific Data* 3: 1–9. https://doi.org/10.1038/sdata.2016.18.